Number Arrangement

DiPS CodeJam 24

Prompt

Given 2 numbers with the same number of digits a and b, can you find whether both follow the same pattern?
For example, take 1462 and 3684.
$4 - 1 = 3, 6 - 4 = 2, 2 - 6 = -4$
$6 - 3 = 3, 8 - 6 = 2, 4 - 8 = -4$
As the differences between digits are the same, they can be said to follow the same pattern.
If the numbers were , say, 1462 and 6738, however:
$4 - 1 = 3, 6 - 4 = 2, 2 - 6 = -4$
$7 - 6 = 1, 3 - 7 = -4, 8 - 3 = 5$
They would not follow the same pattern.
You have a list of n pairs of numbers which you must check to see if they satisfy the given condition. If there are m pairs where the numbers follow the same pattern, what is the sum of all digits of $\sqrt{m}$, if the value is rounded down to the previous integer?

Input Format

- The first line of the input contains an integer n, denoting the number of test cases.
- The next n lines of the input each contain 2 space-separated integers a and b.

Output Format

The first and only line of your output must contain a single integer with the result of your calculations

Constraints

- $10^3 \leq n \leq 10^4$
- $10^8 \leq a, b \leq 10^1 0$

Sample Program

```
def solve(a, b):
  a_digits = [int(i) for i in str(a)]
  b_digits = [int(i) for i in str(b)]

  if len(a_digits) != len(b_digits):
    return False

  a_deltas = [a_digits[i+1]-a_digits[i] for i in range(len(a_digits)-1)]
  b_deltas = [b_digits[i+1]-b_digits[i] for i in range(len(b_digits)-1)]

  return False if a_deltas != b_deltas else True
```